Journal of Computers, Mechanical and Management
Journal homepage: www.jcmm.co.in
e-ISSN: 3009-075X

Volume 4 Issue 2
Article Number: 25214

A Narrative Review of Data Mining Techniques for User Behavior Recognition with Illustrative Application of the Apriori Algorithm

Sonam* and Jyoti

Department of Computer Science Engineering, Baba Mastnath University, Rohtak, Haryana, India 124001

Abstract

This review examines key data mining algorithms used for user behaviour recognition in computational systems, focusing on frequent pattern mining techniques. We summarize foundational methods such as Apriori, FP-Growth, and ECLAT, comparing their operational principles and limitations. A frequency-based literature analysis shows the widespread use of Apriori in market basket analysis. To illustrate its workings, we include a demonstrative walkthrough of the Apriori algorithm using a hypothetical dataset. The article concludes with insights into performance trade-offs and future directions in algorithmic efficiency.

Keywords: Data Mining; User Behavior; Apriori; FP-Growth; ECLAT; Association Rule Mining

1. Introduction

In today's data-driven landscape, organizations across industries—such as retail, healthcare, and finance—generate massive amounts of data daily. The critical challenge lies in converting this raw data into actionable insights, particularly for understanding user behaviour patterns. Data mining, also known as Knowledge Discovery in Databases (KDD), plays a central role in extracting such patterns [1]. Frequent Pattern Mining (FPM), a key subfield of data mining, focuses on discovering recurring relationships among data items. These patterns often indicate significant behavioural traits such as co-purchasing habits or symptom clusters, which are useful for strategic planning and decision-making. Algorithms like Apriori [2, 3], FP-Growth [4], and ECLAT [5] have been widely applied in tasks like market basket analysis to uncover such patterns. This review summarizes major FPM techniques used in behaviour recognition, compares their computational characteristics, and highlights their practical strengths and weaknesses. A step-by-step example using the Apriori algorithm on a hypothetical dataset is included to illustrate its working principles.

2. The Knowledge Discovery Process

Knowledge Discovery in Databases (KDD) is a structured, multistage process that converts raw data into actionable insights. It includes the sequential steps of data selection, preprocessing, transformation, data mining, and interpretation. Each stage plays a vital role—selection targets relevant datasets, preprocessing handles noise and inconsistencies, transformation formats the data for analysis, mining extracts patterns, and interpretation evaluates these patterns for their real-world utility. These steps are especially critical when analyzing user behaviour, as clean, well-structured data is essential for detecting accurate patterns. Figure 1 illustrates the overall KDD pipeline.

*Corresponding Author: Sonam (sonamyadav2706@gmail.com)
Received: 18 Jan 2025; Revised: 25 Apr 2025; Accepted: 26 Apr 2025; Published: 30 Apr 2025

DOI: 10.57159/jcmm.4.2.25214.

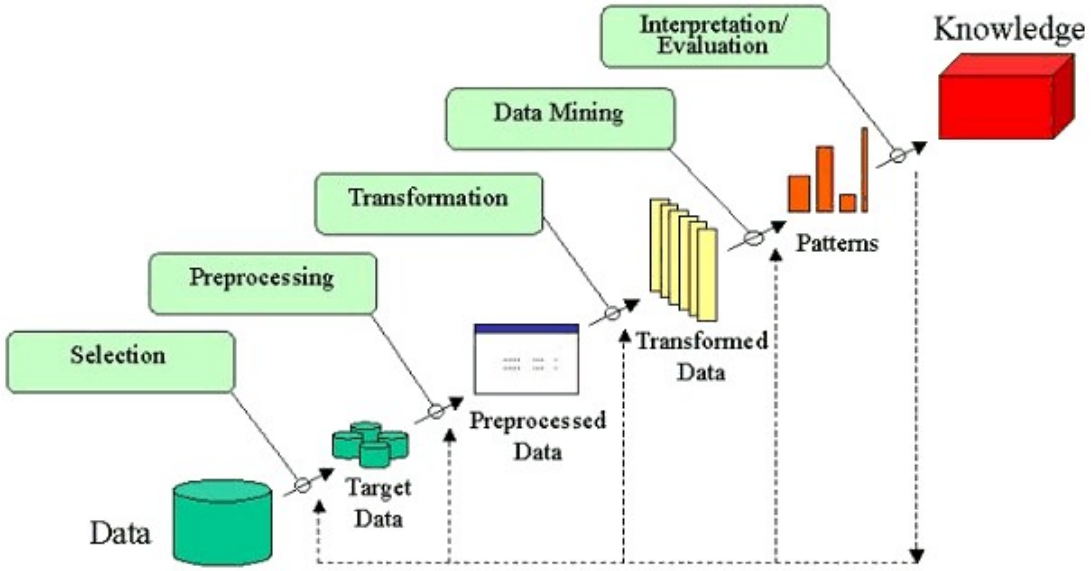

Figure 1: Stages of Knowledge Discovery in Databases (KDD), from data selection to knowledge interpretation [6].

3. Methodology

This study follows a narrative review framework supplemented with an illustrative example. It investigates frequent pattern mining (FPM) algorithms applied to user behaviour recognition, focusing on Apriori, FP-Growth, and ECLAT. A total of 13 academic sources were reviewed, including journal articles and conference proceedings published between 1993 and 2017. The literature was manually gathered from repositories such as IEEE Xplore, SpringerLink, and Google Scholar, prioritizing works that detailed algorithmic performance or application-specific evaluations. To demonstrate practical application, a step-by-step walkthrough of the Apriori algorithm is provided using a small hypothetical dataset. This example illustrates how frequent itemsets and association rules are generated using defined support and confidence thresholds. As a narrative review, this work does not follow a systematic review protocol or include formal quality appraisal. Instead, it aims to synthesize core themes and trends in algorithm design and application. The hypothetical dataset serves as an educational tool and does not reflect the complexity of real-world transactional data.

4. Frequent Pattern Mining Techniques

Frequent Pattern Mining (FPM) is a key technique in data mining for identifying recurring relationships among items in large datasets. One of its most widely known applications is market basket analysis, where analysts identify item combinations that frequently co-occur in purchase transactions [4, 2]. The goal is to discover itemsets that appear together in the dataset with a frequency above a specified threshold, known as support. Once these frequent itemsets are identified, association rules can be generated to describe conditional relationships, typically measured using metrics such as confidence and lift. Several algorithms have been developed to address the computational challenges of frequent itemset mining. The most prominent include:

- **Apriori Algorithm**: Introduced by Agrawal and Srikant in 1994, this foundational method uses a level-wise approach and the Apriori property to eliminate infrequent itemsets early in the process. It requires multiple database scans and is sensitive to high dimensionality [2, 3].
- **FP-Growth**: This algorithm avoids candidate generation by compressing the dataset into a prefix-tree structure called the FP-tree. It then recursively extracts frequent patterns from the tree, resulting in faster performance, particularly for large and sparse datasets [4].
- **ECLAT (Equivalence Class Clustering and bottom-up Lattice Traversal)**: Unlike the above horizontal-format methods, ECLAT uses a vertical data format and applies set intersection on transaction ID lists to compute support. It is often more efficient on dense datasets but can consume more memory [5].

The effectiveness of each algorithm depends on factors such as dataset size, itemset density, and dimensionality. Their relative performance and resource efficiency have been the focus of numerous comparative studies. In practice, frequent pattern mining is implemented within layered data mining systems that integrate user interfaces, data preprocessing modules, mining engines, and result visualization components. These systems structure the transition from raw data to actionable knowledge. Figure 2 illustrates the typical architecture of a data mining system.

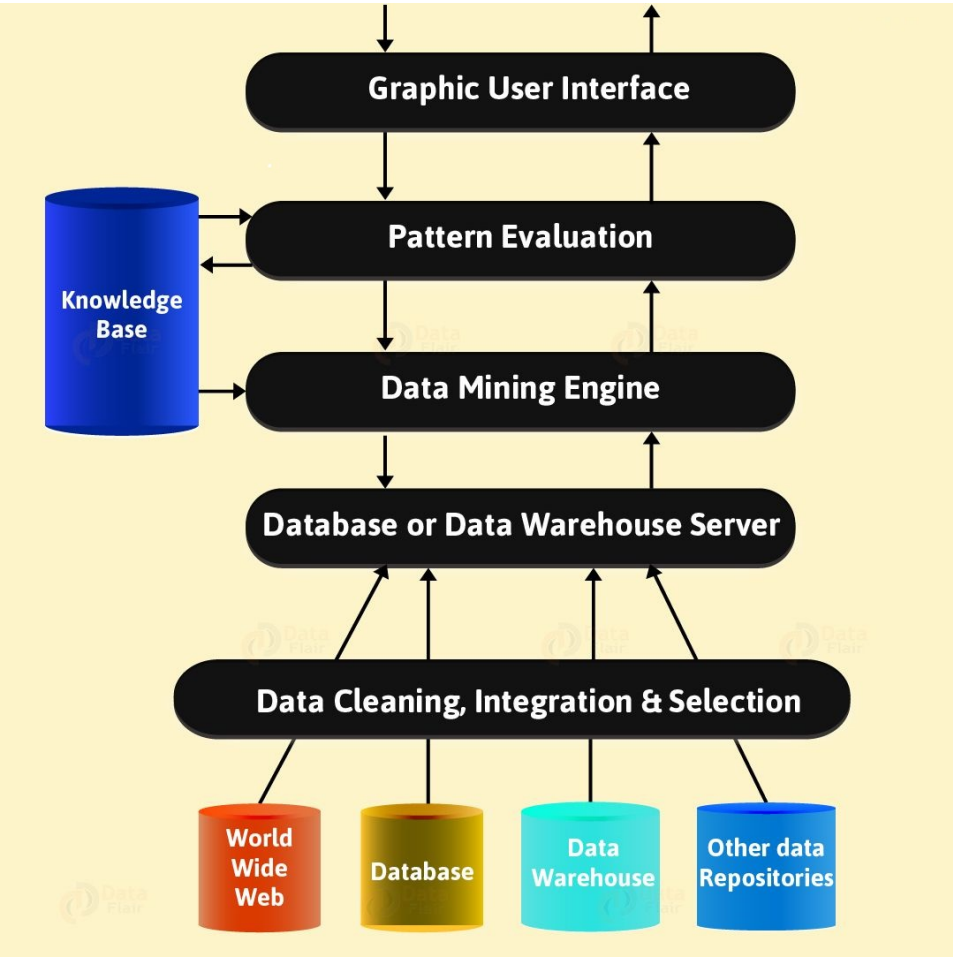

Figure 2: Typical data mining system architecture, showing the flow from user interface to data sources and pattern discovery [6].

FPM algorithms can also be categorized by their underlying computational strategies. Figure 3 shows this classification, which includes join-based methods like Apriori, tree-based methods like FP-Growth, and vertical intersection methods such as ECLAT. This taxonomy highlights key differences in how algorithms process and organize data.

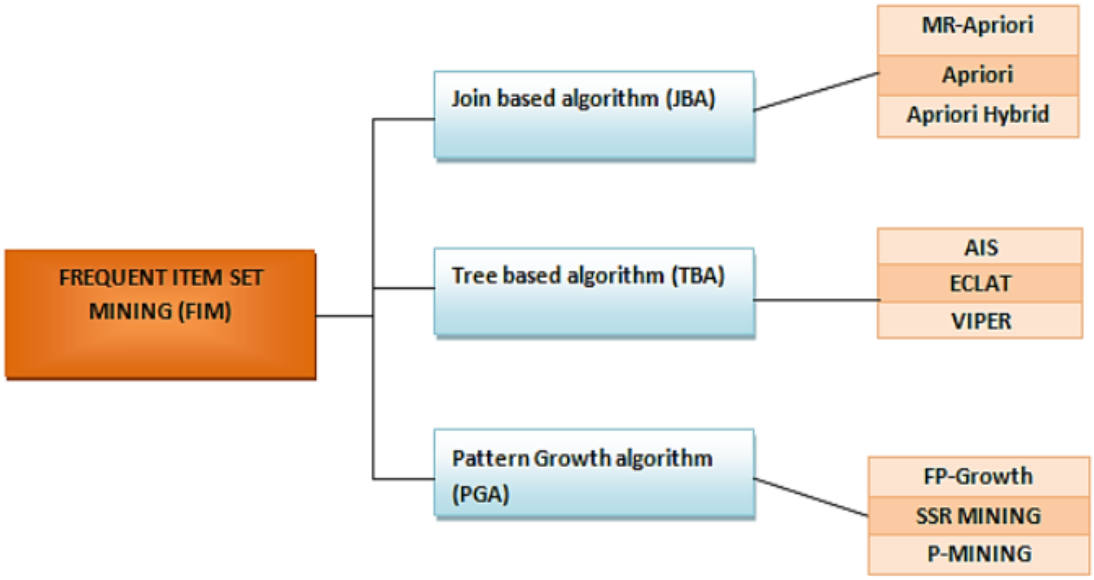

Figure 3: Classification of frequent itemset mining algorithms based on their core strategy: join-based, tree-based, and vertical format methods.

5. Association Rule Mining Applications

Association Rule Mining (ARM) is a data mining technique used to uncover co-occurrence relationships between variables in large transactional datasets. It builds on frequent pattern mining by identifying if-then rules that describe how the presence of one set of items in a transaction implies the presence of another. ARM is widely applied in domains such as retail, healthcare, education, and cybersecurity [4, 7]. In retail, one of the most common applications is market basket analysis. For example, if many customers who purchase bread and milk also purchase eggs, the association rule `Bread, Milk` $\rightarrow$ `Eggs` can be generated. These patterns inform product placement, promotional bundling, and inventory management. In healthcare, ARM is used to detect associations between symptoms, diagnoses, or treatments. A rule might reveal that patients diagnosed with condition A often exhibit symptom B, aiding early diagnosis or treatment planning [7]. Similar applications are found in educational analytics, where patterns in student behavior or performance can guide interventions. ARM relies primarily on three metrics:

- **Support**: The proportion of transactions that contain a given itemset.
- **Confidence**: The conditional probability that a transaction containing one itemset also contains another.
- **Lift**: The ratio of observed support to that expected if the itemsets were independent. A lift greater than 1 indicates a positive association.

These metrics help evaluate the relevance and strength of discovered rules. Together, ARM techniques contribute not only to commercial recommendation systems but also to behaviour modeling, anomaly detection, and strategic decision support across diverse sectors.

6. Apriori Algorithm: Mechanism and Demonstration

The Apriori algorithm is one of the earliest and most widely used algorithms for mining frequent itemsets and association rules. It operates by iteratively identifying frequent itemsets of increasing length, using the principle that all non-empty subsets of a frequent itemset must also be frequent [2, 8]. Apriori employs a breadth-first search strategy. In each iteration, candidate itemsets are generated by joining frequent itemsets from the previous iteration. These candidates are then pruned based on a minimum support threshold. Once all frequent itemsets are identified, association rules are generated and evaluated using confidence thresholds. To demonstrate the algorithm's working, consider a hypothetical dataset of transactions:

Table 1: Sample Transactions

Transaction ID	**Items**
T100	1, 3, 4
T200	2, 3, 5
T300	1, 2, 3, 5
T400	2, 5

With a minimum support threshold of 50% and confidence threshold of 70%, the algorithm proceeds as follows:

- **Step 1: Count support for each item**. Items 1, 2, 3, and 5 meet the 50% support threshold.
- **Step 2: Generate candidate 2-itemsets**. Frequent pairs include {1,3}, {2,3}, {2,5}, and {3,5}.
- **Step 3: Generate candidate 3-itemsets**. Only {2,3,5} meets the support threshold.
- **Step 4: Generate association rules**. For example, the rule {2,3} $\rightarrow$ {5} is evaluated using confidence: Support({2,3,5}) / Support({2,3}) = 0.5 / 0.75 = 66.7%.

Table 2: Frequent Itemsets and Support

Itemset	**Support**
{2, 3, 5}	50%
{1, 3}, {2, 3}, {2, 5}, {3, 5}	50–75%

While effective and interpretable, Apriori has several known limitations:

- Requires multiple database scans, increasing computational cost.
- Generates large numbers of candidate itemsets.
- Performance degrades with dense or high-dimensional data.

These limitations have motivated the development of more efficient algorithms, which are discussed in the next section. The following table summarizes the key findings from 13 reviewed studies. Each entry includes the authors, research focus, algorithm used, main conclusions, and publication year. This table supports the frequency-based citation analysis and highlights recurring observations across various implementations.

Table 3: Summary of Reviewed Studies on Apriori and Its Variants

No.	Author(s)	Algorithm	Conclusion	Year
1	A. Imran and P. Ranjan [9]	Improved Apriori	Costly, but achieves high computation accuracy.	2017
2	A. Imran and P. Ranjan [9]	Apriori	Remains costly with large computation time.	2017
3	Nadeem Ur-Rahman [10]	Data Mining	Takes large execution time.	2017
4	S. Dhanya et al. [11]	MapReduce Apriori	Uses vertical/horizontal layout; slower execution.	2016
5	R. Karthiyayini and J. Jayaprakash [7]	Apriori	Identifies disease efficiently, but performance decreases with more symptoms.	2015
6	Rahul Shukla and A. K. Solanki [1]	Apriori	Costly and time-consuming.	2015
7	P. Prithiviraj and R. Porkodi [4]	Apriori + Others	Apriori takes more time and gives less accuracy.	2015
8	Paresh Tanna and Y. Ghodasara [3]	Apriori	Does not reduce number of scans; time-consuming.	2014
9	Jayshree Jha and Leena Ragha [2]	Improved Apriori	Applies only to educational data; reduces time but lowers performance.	2013
10	K. Geetha and S. K. Mohiddin [8]	Data Mining	High computational load.	2013
11	Z. Farzanyar and N. Cercone [5]	Data Mining (MapReduce)	Handles large data but slow; only extracts data.	2013
12	C. Kaur [12]	Apriori	Suggests online and single-scan variants for future.	2013

7. Challenges and Future Directions

While frequent pattern mining algorithms such as Apriori, FP-Growth, and ECLAT have proven useful in behavioural analysis and recommendation systems, they are not without limitations. Several challenges were consistently noted across the reviewed literature:

- **Scalability**: Algorithms like Apriori perform poorly on large or dense datasets due to repeated scans and exponential candidate growth.
- **Memory consumption**: Storing and evaluating large sets of candidate itemsets or trees (in FP-Growth) can exceed available memory, especially in real-time applications.
- **Runtime complexity**: High-dimensional data leads to longer processing times, making these algorithms less practical in environments with tight latency requirements.

To address these challenges, future work in the field is exploring several directions:

- **Hybrid algorithms**: Combining features of Apriori and FP-Growth, or integrating pruning techniques from different paradigms, can reduce overhead. Early research has shown hybrid approaches to improve speed and memory efficiency [4].

- **Parallel and distributed mining**: Frameworks like Hadoop and Spark have been used to improve runtime on large datasets by distributing workload across nodes [13, 9].
- **Memory-efficient data structures**: Advanced indexing techniques and vertical data layouts can reduce the algorithm's memory footprint, especially in MapReduce contexts [11].
- **Expanded mining scope**: Extending FPM to handle temporal, hierarchical, or streaming data can broaden its applicability in real-time analytics. This is a growing research direction aimed at making mining suitable for dynamic datasets.

Future studies should also aim to benchmark algorithms using real-world datasets and standardized performance metrics such as runtime, memory usage, precision, and scalability. This will help validate theoretical improvements and support more informed selection in practical deployments.

8. Conclusion

This article reviewed major data mining algorithms used in user behaviour recognition, with a focus on frequent pattern mining techniques such as Apriori, FP-Growth, and ECLAT. By analyzing their frequency of use in the literature and discussing their operational mechanisms, we highlighted both their strengths and limitations. A step-by-step demonstration of the Apriori algorithm on a hypothetical dataset illustrated how frequent itemsets and association rules are generated. Among the reviewed techniques, Apriori remains the most cited and widely taught, though it is increasingly challenged by more efficient alternatives in practical settings. The review identifies key performance trade-offs and recurring implementation issues such as runtime complexity and memory use. Future research should continue exploring hybrid models and distributed systems to enhance algorithmic efficiency, especially in large-scale, real-time data environments where existing methods struggle with scalability and resource constraints. Ultimately, the choice of algorithm should be guided by data characteristics, available computational resources, and the specific goals of the behaviour analysis task.

Declaration of Competing Interests

The authors declare no known competing financial interests or personal relationships.

Funding Declaration

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Author Contributions

Sonam: Investigation, Data Curation, Software, Visualization, Writing - Original Draft; **Jyoti**: Supervision, Conceptualization, Methodology, Writing - Review and Editing.

References

[1] R. Shukla and A. K. Solanki, "Performance analysis of frequent pattern mining algorithm using apriori on medical data," *International Research Journal of Computer Science (IRJCS)*, vol. 2, no. 10, 2015.

[2] J. Jha and L. Ragha, "Educational data mining using improved apriori algorithm," *International Journal of Information and Computation Technology*, vol. 3, no. 5, 2013.

[3] P. Tanna and Y. Ghodasara, "Using apriori with weka for frequent pattern mining," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 12, no. 3, 2014.

[4] P. Prithiviraj and R. Porkodi, "A comparative analysis of association rule mining algorithms in data mining: A study," *American Journal of Computer Science and Engineering Survey*, vol. 3, no. 1, pp. 98–119, 2015.

[5] Z. Farzanyar and N. Cercone, "Efficient mining of frequent itemsets in social network data based on mapreduce framework," in *Proc. IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 1183–1188, 2013.

[6] DataFlair, "Data mining architecture - components, types & working." https://data-flair.training/blogs/data-mining-architecture/, n.d.

[7] R. Karthiyayini and J. Jayaprakash, "Association technique on prediction of chronic diseases using apriori algorithm," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 4, May 2015. Special Issue 6.

[8] K. Geetha and S. K. Mohiddin, "An efficient data mining technique for generating frequent itemsets," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 4, pp. 571–575, 2013.

[9] A. Imran and P. Ranjan, "Improved apriori algorithm using power set on hadoop," in *Proc. 1st International Conference on Computational Intelligence and Informatics*, (Hyderabad, India), pp. 245–254, 2017.

[10] N. Ur-Rahman, "Textual data mining for knowledge discovery and data classification: A comparative study," *European Scientific Journal*, vol. 13, no. 21, 2017.

[11] M. Dhanya, M. Vysaakan, and A. Mahesh, "An enhancement of the mapreduce apriori algorithm using vertical data layout and set theory concept of intersection," in *Intelligent Systems Technologies and Applications*, vol. 385, pp. 225–233, 2016.

[12] C. Kaur, "Association rule mining using apriori algorithm: A survey," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, no. 6, 2013.

[13] Y. Rochd and I. Hafidi, "An enhanced apriori algorithm using hybrid data layout based on hadoop for big data processing," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 18, no. 6, 2018.